METHOD AND SYSTEM FOR CAPTURING, STORING AND RETRIEVING EVENTS AND ACTIVITIES

5 **FIELD OF THE INVENTION**

This invention relates to a method and apparatus for processing event and activity data of a process. In particular, the method and apparatus of the present invention is concerned with processing event and activity data of any type and number of systems.

10

BACKGROUND OF THE INVENTION

A process takes place over a period of time. During the process various events and activities occur and various parameters vary in value. There is a
5   need to monitor a process in order to analyze its performance or of any parameters thereof, whether the process is an industrial one for the handling, treatment or flow of material or other process, such as the tracking of the weather or of commodities or other financial instruments and the like.

10   An event is something that happens at a specific point in time, e.g., an alarm becoming active, an operator message being confirmed, a parameter download, recordation of a lab measurement, and the like. An activity is something that happens over a period of time, e.g., a movement of material, a batch, a phase inside a batch, an alarm being in an active state, and the like.
15   An activity has a time span, interval or period. In contrast, an event occurs at a specific point in time.

In the operational environment of a process, large quantities of historical data can be generated by different data sources. Capturing, storing and making
20   this data available to applications for viewing, analysis and reporting is a challenge. Prior solutions for event and activity data are partial at best. For event and activity data, this problem has mostly been addressed on a system (data source) by system basis, each system supporting a well defined, limited set of event and activity types.
25

Capturing event and activity data is a challenge because the data that needs to be captured for each event and/or activity depends on the event and activity type. For each event and activity type, a different set of (attribute) values may have to be recorded. Another challenge is the fact that the data can come

2

from any number of data sources. Storing the data is a challenge because each data source tends to come with its own solution for historizing the data, which makes the administration task of archiving, recovering and distributing the data very difficult.

5

Making the data available is a challenge because of the multitude of event and activity types that need to be handled and the need to be able to handle new event and activity types by simple configuration, without requiring a new release of the product. Traditionally, different visualization tools have been used

10   for limited, well defined sets of events. The challenge is to combine all this data into a single stream that can be displayed in a single view.

Thus, there is a need for a flexible and efficient method and system for processing event and activity data of any type and any number of systems.

15

SUMMARY OF THE INVENTION

The method of the present invention processes data of a process in a manner that can be used for a variety of different processes without a custom design effort for each process. The method identifies one or more events and/or

5   one or more activities of the process. The method also identifies one or more attributes of each event and/or activity. A generic data structure is used to classify the events, activities and attributes. The data structure comprises an event and/or activity type and a plurality of attribute types. The classification procedure provides defined event and/or activity types for the events and

10   activities and defined attribute types for the attributes. One or more storage volumes are allocated to each of the defined event and/or activity types for storage and retrieval of the process data by attribute type.

According to one aspect of the invention, at least one storage volume is

15   allocated to each of the defined attribute types. According to another aspect of the invention, the data structure further comprises a time stamp. The at least one storage volume is accessed according to the time stamp of an event for storage and retrieval of the data of the attributes thereof.

20   According to another aspect of the invention, the attributes of a plurality of the events and/or activities are common to one of the defined attribute types. One storage volume is allocated to all of the common attributes.

According to another aspect of the invention, the data storage in a

25   storage volume is compressed according to identity of the values of common attributes of consecutive events and/or activities. According to another aspect of the invention, the data structure further comprises a time stamp. The storage volume is accessed according to the time stamp for storage and retrieval of the values. The values of a first event are retrieved from the storage volume by

4

using the value of a first time stamp for the first event or of a second time stamp value of a second one of the events that is earlier in time than the first time stamp value.

5   According to another aspect of the invention, for the case where the values of the attributes of a first defined attribute type are static, the data storage is compressed by omitting storage of a static value in the attribute volume.

10   According to another aspect of the invention, the same generic data structure is used for the definition of event types, activity types and attribute types of each of a plurality of processes, where some are different than others.

15   According to still another aspect of the invention, the data values of events and/or activities that are defined as different event and/or activity types are presented in a single view in any one of a plurality of formats. These formats can be selected from the group consisting of: row format, column format and chart format.

BRIEF DESCRIPTION OF THE DRAWINGS

Other and further objects, advantages and features of the present invention will be understood by reference to the following specification in conjunction with the accompanying drawings, in which like reference characters denote like elements of structure and:

Fig. 1 is a block diagram of a system of the present invention for processing time series data;

Fig. 2 is a block diagram of the computer of the Fig. 1 system;

Fig. 3 depicts an exemplary process, its events, time series data and activities;

Fig. 4 depicts an activity diagram for the Fig. 3 process;

Fig. 5 depicts a data and event diagram for the Fig. 3 process;

Fig. 6 depicts a data structure for the process data handling program of the Fig. 2 computer;

Fig. 7 depicts a partial data history of the process of Fig. 3;

Fig. 8 depicts instances of activities and events;

Figs. 9-11 depict different presentation formats of the process data;

Fig. 12 is a graph of the Fig. 11 data presentation; and

Figs. 13 and 14 are flow diagrams of the process data handling program of the computer of Fig. 2.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring to Fig. 1, a system 20 of the present invention includes a computer 22, a monitor 24 and a database 24. A network 30 interconnects computer 22 and database 26 as well as a client device 32. Monitor 24 monitors

5   a process 28 and provides process data to computer 22. Computer 22 processes the data and stores the data in a memory, such as database 26. Computer 22 may communicate with database 26 via network 30 or directly, as shown by the dashed line 34.

10   Database 26 may be a part of the memory of computer 22 or a separate database, as shown in Fig. 1. Computer 22 may be a single computer or a plurality of computers interconnected via network 30. Network 30 may be any suitable wired or wireless communication network and may include the Internet an Intranet, the public telephone system and the like.

15

Client device 32 may be any suitable computer entry device with a capability to communicate with computer 22 via network 30. For example, if network 30 is the Internet, client device 32 has a browser capability for Internet communications. As such, client device 32 may be a personal computer (PC), a

20   workstation, a phone or other suitable device. Similarly, computer 22 would be equipped with Internet capability to serve files and/or otherwise communicate via the Internet.

Referring to Fig. 2, computer 22 includes a processor 34, a

25   communications unit 36, a memory 38 and a bus 40. Bus 40 interconnects processor 34, communications unit 36 and memory 38. Memory 38 includes an operating system 42 and a process data handling program 44. Operating system 42 controls processor 34 to execute process data handling program 44 to process data of process 28 monitored by monitor 24. A memory media 46

(e.g., a disk) contains a copy of operating system 42, process data handling program 44 or other software, which can be loaded into memory 38. Communications unit 36 includes the capability to communicate via network 30.

5   Process data handling program 44, when run, permits a client to operate client device 32 to identify process 28 in terms of events, time variable parameters and activities. An event is something that happens at a specific time, for example, the triggering of an alarm. Time series data is continuous data of a time variable parameter, such as temperature, pressure, flow rate and 10   the like. An activity is a time interval of the process, for example, the operation of a pump during the process.

       Process data handling program 44, when run, identifies event data and activity data and classifies it according to a generic data structure for storage 15   and retrieval based on the identified event or activity and/or attributes thereof.

       For the purpose of describing the apparatus and method of the invention, an exemplary process that unloads a material, such as oil, from a ship will be initially described. It is understood, of course, that the system and method of 20   the invention can be used with any process that has events, time variable parameters and/or activities.

       Referring to Fig. 3, a system 50 is shown for running process 28 for pumping out material from a ship 52 into a pair of holding tanks, Tank 1 and 25   Tank 2. Ship 52 has a level indicator LI001 that monitors the material level in ship 52 during pump out process 28. A temperature monitor T101 monitors the outside ambient temperature as it can affect pumping performance. Material is pumped from ship 52 through pipes 54 and passes by a density analyzer A1001. The material is pumped to Tank 1 and Tank 2 by a pair of pumps, P101 and

P102. When a valve V101 is open, the material is pumped by pump P101.
When a valve V102 is open, the material is pumped by pump P102. A valve
V103 controls the flow of material to Tank 1 and a valve V104 controls the flow
of material to Tank 2. Flow rate to tanks using pump P101 is monitored and
5   controlled a flow analyzer FI1001. Flow rate to tanks using pump P102 is
monitored and controlled by a flow analyzer FI1002. A level indicator LI101
monitors the level of material in Tank 1 and a level indicator LI102 monitors the
level of material in Tank 2. A motor M101 controls an agitator 54 in Tank 1 and
a motor M102 controls an agitator 56 in Tank 2.

10   In system 50, the following constraints apply:

        a. Contents of ship 52 do not fit into a single tank.
        b. Only one tank can be filled at a time.
15   c. Only one pump can be used at a time.

    Referring to Fig. 4, a number of activities are defined for the pump out
process of ship 52. The execution of the activities begins at a time T0 and
completes at a time Tn. These activities are as follows:
20

    1) Analyze activity – This activity continuously monitors analyzer A1001
       throughout the pump out process from time T0 to time Tn and alerts an
       operator if the material density is outside of a specified range.

25 2) Unload Ship activity – This activity is a procedure for initiating and
       monitoring the pump out of ship 52. It is the master procedure
       responsible for initiating sub activities: Pumpout1, Pumpout2, Pumpout3,
       Mix1, and Mix2.

3) Pumpout1, Pumpout2, Pumpout3 activities – These three pump out activities are of the same type, but each uses different settings. A pump out activity stops when a tank is full or a failure (e.g. pump failure) occurs. In such case, the higher-level activity Unload Ship is responsible to schedule another pump out activity with the appropriate settings until ship 52 is empty. Activity Pumpout1 moves material from ship 52 to Tank 1. Activity Pumpout1 stops when Tank 1 is full. Activity Pumpout2 pumps material from ship 52 to Tank 2. Activity Pumpout2 stops when pump P101 fails. Activity Pumpout3 uses pump P102 and continues pumping material to Tank 2 until ship 52 is empty.

4) Mix 1 activity – This activity is a procedure responsible for activating agitator 54 for Tank 1. Agitator 54 starts during activity Pumpout1 and runs for a period of time after the completion of activity Pumpout1.

5) Mix 2 activity – This activity is a procedure responsible for activating agitator 56 for Tank 2. Agitator 56 starts during activity Pumpout2, continues during activity Pumpout3 and runs for a period of time after the completion of activity Pumpout2.

These activities can be expressed in a hierarchical order of activity, sub-activity and sub-sub-activity as shown in Table 1.

TABLE 1

Activity	Unload Ship
Sub-activity	Pumpout1
Sub-sub-activity	Mix1
Sub-activity	Pumpout2
Sub-sub-activity	Mix2

Sub-activity	Pumpout3
Sub-sub-activity	Mix2

Referring to Fig. 5, a curve 60 represents the material level in ship 52 (i.e., the values of level sensor LI001) during the activity Unload Ship. Curve 62 represents the material level in Tank 1 (i.e., the values of level sensor LI101)

5   during the sub-activity Pumpout1. Curve 64 represents the material level in Tank 2 (i.e., the values of level sensor LI102) during sub-activity Pumpout2 and sub-activity Pumpout3. A curve 66 represents the ambient temperature (i.e., the values of temperature sensor T101) during the activity Unload ship. A curve 68 represents the flow rate of material as monitored by flow rate sensor FI1001

10   during sub-activity Pumpout1. A curve 70 represents the flow rate of material as monitored by flow rate sensor FI1001 during sub-activity Pumpout2. A curve 72 represents the flow rate of material as monitored by flow rate sensor FI1002 during sub-activity Pumpout3. As can be seen, the outputs of level sensors LI001, LI101 and LI102 and of flow rate sensors FI1001 and FI1002 vary with

15   time and are continuous or time series data.

Figs. 3-5 show the execution of the activities and sub-activities required to pump out ship 52 over a period of time. At time T0, an instance of activity Unload ship is created. The operator would give the instance a name, for

20   example, UNLOAD_2001_06_01. The operator would then initiate the activities in either an automated or manual manner.

The process data shows a plurality of events 74, 75, 76, 77, 78 and 79 that occur during the pump out process. Event 74 represents a flow change

25   initiated by the operator to increase the flow rate during sub-activity Pumpout1. This flow rate change is monitored by sensor FI1001. Event 76 represents a temperature alarm detected when the ambient air temperature drops below a

safe pump operating range during sub-activity Pumpout2. Event 78 represents a failure of pump P101 during sub-activity Pumpout2. Event 75 represents a failure off of pump P101 at the end of activity PumpOut2. Event 79 represents a red tag out of service at the start of the pump 1 repair activity. Event 77

5   represents a red tag in service at the end of the pump 1 repair activity. .As a result of the failure of pump P101, the process switches to the second pump P102.

Process 28 is initially defined according to a generic data structure that

10   has a data framework for activities and events and attributes of each. The data structure is generic in that it is useable for a plurality of different processes, so that a totally new custom design for each process is unnecessary.

Process data handling program 44 identifies activities and attributes of

15   each for process 28. The identification step is performed in response to an interactive session with a user operating client device 32. Alternatively, events and activities may be identified by process data handling program 44 as data from process 28 is gathered.

20   Referring to Fig. 6, a data structure 80 includes a framework 82 for activities and a framework 84 for events. Framework 82 defines an activity by type, attributes and characteristics of the attributes, as shown in boldface type. Framework 84 defines an event by type, attributes and characteristics of the attributes, as shown in boldface type. The characteristics of frameworks 82 and

25   84, e.g., define whether the attribute is static or dynamic (variable) or whether the attribute has a default value. It will be apparent to those skilled in the art that frameworks 82 and 84 may include other characteristics.

Process data handling program 44 classifies the events and activities of process 28 and their attributes identified by the identification step according to frameworks 82 and 84 of data structure 80. Thus, the unit operation activity type, the alarm event type and the operator change event type are shown in Fig.

5   6 as classified within data frameworks 82 and 84. The unit operation activity type has defined attributes of start time, end time, identification (ID), equipment, flow rate, accumulated value (AccumValue) and source. The alarm event type has defined attributes of time stamp, resource, message, state and source. The operator change event type has defined attributes of time stamp, operator,

10 resource, message, command and source.

Referring to Fig. 7, a history 86 of the data of process 28 is shown for three PumpOut activities, three alarm events and three operator change events. The PumpOut activities are PumpOut1, PumpOut2 and PumpOut3. For example, the data history for activity PumpOut1 has a start time of 8/17/200 at

15 10:00AM, and end time of 8/17/2000 at 10:55AM and an equipment of Tank1.

There is one alarm event for temperature sensor T101 and two alarm events for pump P101. For example, the alarm event for temperature sensor

20 T101 has a time stamp of 8/17/2000 at 11:35AM, a message of low temperature and a state of ON.

There is one operator change event for a resource, pump P101 that has a time stamp of 8/17/2000 at 10:20AM, a message of Speed and an operator

25 named Joe. There are also two operator change events for a device, PMP101. Process 28 includes various systems that use diverse external names, e.g., device or resource, for the same field or attribute name. Also, the diverse systems may also use different field contents for the same operational mechanism. In this case, pump P101 and pump PMP101 are the same pump.

Process data handling program 44 includes a global dictionary or map structure that is used to map or transform (1) external field names, such as resource and device, to an internal field name or attribute of resource and (2)

5   external field contents, such as P101 and PMP101, to internal attribute fields of P101.

Referring to Figs. 6 and 8, process data handling program 44 organizes storage of event and activity data by event type and activity type and by attribute

10   thereof. Thus, the data for all of the PumpOut activities, the alarm events and the operator change events is stored by attribute. For example, the start times of PumpOut1, PumpOut2 and PumpOut3 are all stored in one logical or physical storage volume and the end times thereof are stored in another logical or physical storage volume. Thus, process data handling program 44 converts a

15   request to store or retrieve a start time for activity PumpOut1 to a form that accesses memory 38 for the storage volume that is allocated for the start times of the PumpOut activities, specifically the location therein for the start time of activity Pumpout1.

20   In a preferred embodiment of the invention, storage is optimized by using compression. If a current value is identical to the last stored value of an attribute type, then the current value does not need to be stored. This is called equality compression. The double border boxes in Fig. 8 show some current values that need not be stored. For example, box 88 shows a current value of Tank2, which

25   is a repeat of the last stored value of the equipment attribute type of the Pump Out activities. As another example, box 90 shows a current value of ON, which is a repeat of the last stored value of the state attribute type of alarm events.

If a value of an attribute type always has the same value for a specific event and/or activity type, optimization is achieved by defining it as a static attribute and the storing the attribute value as a part of the event and/or activity type definition. For example, boxes 92 in Fig. 8 show a static attribute value of
5   operating system. These values need not be stored.

These optimization rules or policy can be summarized as follows:

1. Store each attribute in a particular storage volume.

10

2. Do not store an attribute value if it is identical to the preceding value in the storage volume.

3. When retrieving the attribute, use the value for the timestamp
15   of the event (or activity) or the preceding value.

Referring to Figs. 9-11, data of different event and/or activity types can be retrieved and presented in any one of a plurality of formats. In Fig. 9, the data is presented in a row format in which the attributes of an event (or activity) are
20   presented in a row. This format is useful for generic reports. In Fig. 10, the data is presented in a column format in which the event (or activity) types are presented in rows and the attribute values are presented in columns. This format is useful when looking for specific event or activity types. In Fig. 11, the data is presented in a time series format in which the value for a particular event
25   (or activity) is taken for a number of events (or activities). This format is extremely useful for generating graphical views.

Referring to Fig. 12, the accumulated values of the Fig. 11 presentation are plotted against the PumpOut end times.

16

Referring to Fig. 13, process data handling program 44 permits a user to define a process in terms of data structure 80. Step 100 identifies the activities, sub-activities, sub-sub-activities and so on of the process. Step 102 identifies

5   activity type attributes, such as start times and end times, time variable parameters (e.g., flow rate), equipment and tags. Step 104 classifies the activities defined by step 100 and the attributes defined by step 102 according to data structure 80. Step 104 also interprets these activities and attributes to build the global dictionary maps required to map diverse (but equivalent) external

10 names and field contents to a common internal field name and field content.

Step 106 identifies event types. Step 108 identifies event type attributes. Step 110 classifies event types defined by step 106 and the attributes defined by step 108 according to data structure 80. Step 110 also interprets these events and attributes to build the global dictionary maps required to map diverse (but

15 equivalent) external names and field contents to a common internal field name and field content. Step 112 identifies tags for sensors that monitor time series data. Step 114 accepts the tags defined by step 112. Step 116 stores the definitional data accepted by steps 104, 110 and 114 in data base 26 in a manner that permits access by activity, event, attributes of either and/or tags.

20 For example, database 26 may be physically or logically organized by activity, event and attributes of either. If logically organized, a storage access translator would be used to translate the activity, event, attributes of either and/or tag access data into physical storage volumes.

25

Referring to Fig. 14, process data handling program 44 also permits the collection of data during the running of the process, such as process 28. Step 120 identifies that the process has been initiated by the operator or automatically by a control system. Step 122 executes activities, such as

17

Pumpout1. Step 124 determines when step 122 is finished. Step 130 closes the collection of activity data.

Step 126 creates an activity history record, such as attribute values (e.g.,
5   start time). Step 128 collects event happenings, time stamps and the like for events. Step 130 closes an activity history record, such as attribute values (end time). Step 132 processes the event happenings and links them to activities of any tier. Step 134 collects time series data monitored by the various sensors of the process.

10

Step 136 stores the activity, event and time series data in database 26 for retrieval by activity, attribute thereof and/or tag. Step 138 retrieves the data activity, event, attribute and/or sensor tag for processing or analysis.

15   The present invention having been thus described with particular reference to the preferred forms thereof, it will be obvious that various changes and modifications may be made therein without departing from the spirit and scope of the present invention as defined in the appended claims.